END
DATE
FILMED
10 80
DTIC

|||| 1.0

45
50
56

2.8

2.5

3.2

2.2

36

40

2.0

|||| 1.1

1.8

||| 1.25

1.4

1.6
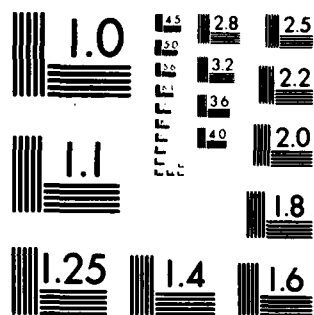
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

LEVEL II
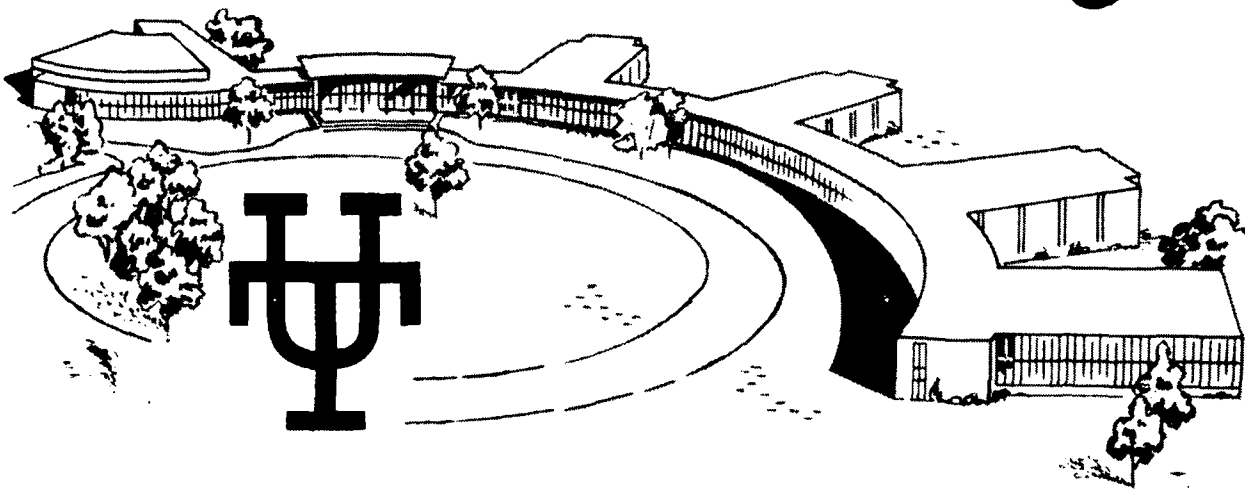
(10)

DTIC
ELECTE
SEP 2 8 1980
C

80 9 25 061

THE UNIVERSITY of TENNESSEE
SPACE INSTITUTE

Tullahoma, Tennessee

18 19

AFOSR-TR-80-0753

6

CONTROL OPTIMIZATION FOR A DUAL-MODE
SINGLE-STAGE NUCLEAR SHUTTLE

10

Thomas Conley Powell

Susan Gay Dodd

University of Tennessee Space Institute

Tullahoma, Tennessee 37388

ABSTRACT

The performance of a single-stage surface-to-orbit shuttle--
whether chemical-propellant or nuclear--can be considerably improved by
mixed-mode propulsion. A mixed-mode shuttle would be fitted with
engines designed to use two different propellant combinations--a high-
thrust (mode 1) propellant, such as ammonia, and a high-specific-impulse
(mode 2) propellant, such as hydrogen.

The first step in the evaluation of the mixed-mode nuclear shuttle
is a preliminary trajectory optimization study. This study, using a
simple mission and a simple shuttle model, would be the basis for more
complex trajectory optimization studies. The problem considered in this
thesis is that of minimizing the propellant expenditure of a mixed-mode
nuclear shuttle for a given orbit. The starting point for any optimiza-
tion problem is a mathematical model of the system, in state variable
form.

Application of the methods of optimal control theory results in a
two-point boundary-value problem and an associated algebraic problem.
Three numerical methods are briefly described which could be used to
solve these problems. However a more flexible method, that of finite
differences, is proposed in this study to handle the more restricted
problem. Replacement of the derivatives with finite-difference approxi-
mations results in a set of nonlinear algebraic equations. These
equations are solved by the Newton-Raphson technique; central-difference
approximations are used for the Jacobian matrix, and the linearized
algebraic equations are solved by the Gauss elimination method. A
procedure is presented for obtaining good starting values for the

i

Newton-Raphson iteration. This is done by neglecting the effects of atmosphere; as a result, the optimality conditions are greatly simplified. As it turned out, this procedure proved sufficient for an evaluation of mixed-mode propulsion. Consequently, the finite-difference method was not used to obtain numerical results, although the correctness of the program was verified.

The analysis of the problem is given in detail, along with numerical results for various combinations of the input variables, and the finite-difference computer program, properly documented.

In conclusion, suggestions for extensions of this work are presented and methods of approach are outlined.

Accession For		
NTIS GRAAI	☑	
DOC TAB	☐	
Unannounced	☐	
Justification___		
By___		
Distribution/		
Availability Codes		
Dist.	Avail and/or special	
A		

TABLE OF CONTENTS

LIST OF TABLES

CHAPTER I

INTRODUCTION

Since the end of the Apollo program, the trend in space transportation has been towards low cost. The intention was that reusability of space vehicles and equipment, commonality, and integration of manned and unmanned flights would accomplish this task. The space shuttle was born out of this trend.

The present shuttle was supposed to drastically reduce the cost of surface-to-orbit transportation. However, this shuttle will be cheaper to develop than to operate. The estimated payload costs were 350 dollars per kilogram per low altitude low inclination orbits and the actual figure will undoubtedly be substantially higher.

The present shuttle has operational deficiencies as well. One of its main objectives is to carry large payloads into earth orbit with greater reliability and economic payback. But the payload capability itself is quite limited--only 29,500 kilograms maximum in low altitude low inclination orbit and this decreases rapidly as orbital altitude and inclination increase. In addition, there is very little orbital maneuver capability. Finally, one has the nuisance of sea-recovering the boosters.

Therefore it is logical to look to the possibility of a single-stage shuttle because operational and developmental costs could be greatly reduced. But the present propulsion systems do not permit single-stage winged shuttles with reasonably large payload fractions or perhaps with any payload at all.

The concept of mixed-mode propulsion may make a reusable one-stage-to-orbit shuttle feasible. "Mixed-mode" propulsion for surface-to-orbit

shuttles was first proposed by Robert Salkeld in an article published in 1971.[1] There is a tradeoff between thrust and specific impulse, but this tradeoff shifts during ascent. High thrust is at a premium at liftoff, but as speed and altitude are built up the tradeoff shifts in the direction of high specific impulse. A mixed-mode shuttle would be fitted with engines designed to burn two different propellant combinations--a high-thrust (mode 1) combination, such as kerosene and oxygen, and a high-specific-impulse (mode 2) combination, such as hydrogen and oxygen--simultaneously. At liftoff, the engines would be operating in mode 1; they would transition to mode 2 during ascent. Detailed studies have shown that mixed-mode propulsion makes single-stage winged shuttles of modest size practical.

Based on initial economic analysis, the initial cost of mixed-mode shuttles will be far less than for the single-mode alternatives. Robert Salkeld believes the mixed-mode concept can cut the overall cost of a 1000-flight shuttle program at least 40% while producing a much more effective system.

Operationally, the mixed-mode shuttle also seems to be a good bet. The mixed-mode propulsion can increase performance in a vehicle of a given size or reduce the size of a vehicle of a given performance. Propellant volume is reduced, allowing for a larger payload capability, thus satisfying one of the main objectives for the space shuttle. Thus the potential performance gains from mixed-mode propulsion warrant its study.

Donald Kingsbury has pointed out that the mixed-mode concept can also be applied to nuclear rockets.[2] The most likely mode 1 propellant is ammonia, which has a fairly high density but readily

dissociates into a mixture of nitrogen and hydrogen having a fairly low molecular weight. The mode 2 propellant would be hydrogen.

The first step in the evaluation of the performance of a mixed-mode nuclear shuttle is an ascent-trajectory optimization study. The purpose of this work is to program the thrust vector for a mixed-mode single-stage nuclear surface-to-orbit shuttle so as to minimize the propellant expenditure required to reach a given orbit. (Minimizing the propellant expenditure required to reach a given orbit is equivalent to maximizing the final mass of the shuttle.) The following assumptions are made:

1. The shuttle is launched from a site on the equator into an equatorial orbit. As is appropriate for a preliminary study, this assumption is restrictive, but not unrealistic. It considerably simplifies the computation and the presentation of results, without sacrificing any of the essential features of the problem. The shuttle's performance would be slightly reduced for non-equatorial launch sites and non-equatorial orbits.

2. The nozzle flow may be treated as a one-dimensional ideal-gas flow. This assumption is acceptable for a preliminary study, and it greatly simplifies the computation.

3. The nozzle flow is expanded to ambient pressure. This assumption implies that the shuttle is fitted with an aerospike nozzle. If this is not the case, the assumption is slightly optimistic.

4. The effective pump power and the reactor-exit stagnation temperature are independent of the reactor-exit stagnation pressure. ("Effective pump power" is the power required to pump the propellants, assumed to be incompressible, from zero pressure to the reactor-exit stagnation pressure.) The accuracy of this

assumption depends upon the quality of the pump and heat-
exchanger designs.

5. The effect of Reynolds number upon the drag and lift coeffi-
    cients is negligible.

6. Atmospheric temperatures and pressures are approximately those
    of the Standard Tropical Atmosphere.

The following are given, in addition to the configuration and the
choice of propellants:

1. Final altitude and velocity.

2. Reactor-exit stagnation temperature.

3. Effective pump power/initial mass.

4. Nozzle throat area/initial mass.

5. Base area/initial mass.

(Note that, for a given family of designs, the payload/initial mass ratio
may depend upon all these things, as well as upon the propellant capacity
per unit initial mass.)

The following is an outline of the remainder of this work:

CHAPTER II. THE MATHEMATICAL FORMULATION OF THE PROBLEM

The differential equations (equations of motion) which describe the
system are derived and explained. Boundary conditions are given.
The performance index is defined.

CHAPTER III. THE OPTIMAL CONTROL PROBLEM

Optimal control theory is applied to obtain the equations which must
be solved to compute the optimal trajectory.

CHAPTER IV. THE NUMERICAL SOLUTION

A finite-difference method for the solution of the optimal control
problem is described. Three alternative numerical methods are also

CHAPTER V.   APPROXIMATE COMPUTATION OF THE OPTIMAL TRAJECTORY

The algorithm for obtaining good starting values to be used to begin the Newton-Raphson iteration.

CHAPTER VI.   NUMERICAL RESULTS AND CONCLUSIONS

The results are presented from the approximate trajectory method.

CHAPTER VII.   RECOMMENDATIONS

Possible extensions of this work and methods of approach are suggested.

CHAPTER II

THE MATHEMATICAL FORMULATION OF THE PROBLEM

The starting point for an optimal control investigation is a mathematical model in state variable form. A state vector for a system is a minimum set of data required to predict the future behavior of the system. The state variables satisfy a set of first-order ordinary differential equations called the state differential equations. The state differential equations contain certain variables in addition to the state variables. These are referred to as the control variables. The control variables are usually taken to be quantities which can actually be manipulated.

The choice of what variables are to be treated as state variables and what variables are to be treated as control variables is purely one of convenience. However, the number of state variables and the number of control variables is determined by the mathematical model of the system.

The control optimization problem is that of finding a control vector that "optimizes" the performance of the system.

For this case, let the state variables be r, $\dot{r}$, and $\dot{\phi}$, where r is the distance from the center of the planet and ϕ is longitude; let the control variables be β and p_r, where β is the angle from the horizontal to the thrust axis, positive upward, and p_r is the stagnation pressure at the reactor exit; and let the independent variable be m, where m is the ratio of the shuttle's instantaneous mass to the initial mass.

The shuttle's radial and transverse accelerations in terms of the state variables and their derivatives are

$$a_r = \ddot{r} - r\dot{\phi} = (F_r + L_r + D_r - mg)/m, \tag{2.1}$$

$$a_\phi = r\ddot{\phi} + 2\dot{r}\dot{\phi} = (F_\phi + L_\phi + D_\phi)/m, \tag{2.2}$$

where F_r, L_r, and D_r are the radial components of the thrust, lift, and drag per unit initial mass and g is the gravitational acceleration and F_ϕ, L_ϕ, and D_ϕ are the transverse components of the thrust, lift, and drag per unit initial mass.

The transverse force is thus

$(F - D)\cos\beta - L\sin\beta,$

and the radial force is

$(F - D)\sin\beta + L\cos\beta - mg,$

where $g = g_0 r_0^2/r^2$, g_0 being the gravitational acceleration at the surface of the planet, and r_0 being the radius of the planet.

Let $(r,\dot{r},\dot{\phi}) = (x_1,x_2,x_3)$, the state variable vector, and let $(p_r,\beta) = (u_1,u_2)$, the control variable vector. The state differential equations are

$$dr/dm = r/(\dot{m}_1 + \dot{m}_2) = f_1(x,u,m), \tag{2.3}$$

$$d\dot{r}/dm = (-g_0 r_0^2/r^2 + r\dot{\phi}^2 + ((F - D)\sin\beta + L\cos\beta)/m)/(\dot{m}_1 + \dot{m}_2) =$$

$$f_2(x,u,m), \tag{2.4}$$

$$d\phi/dm = (-2\dot{r}\dot{\phi} + ((F - D)\cos\beta - L\sin\beta)/m)/(\dot{m}_1 + \dot{m}_2)r = f_3(x,u,m); \quad (2.5)$$

where $-\dot{m}_1$ and $-\dot{m}_2$ are the mode-1 and mode-2 propellant flow rates per unit initial mass, g_0 is the gravitational acceleration at the surface of the planet, r_0 is the radius of the planet, and F, D, and L are thrust, drag, and lift per unit initial mass. The right-hand sides of these differential equations are functions of the state variables, one of the control variables, the independent variable, and $\dot{m}_1$, $\dot{m}_2$, F, D, and L. One must express $\dot{m}_1$, $\dot{m}_2$, F, D, and L as functions of the state and control variables.

The thrust is given by

$$F = -(2\gamma_e RT_r/(\gamma_e - 1)W_e)^{1/2}(1 - (p_a/p_r)^{(\gamma_e - 1)/\gamma_e})^{1/2}(\dot{m}_1 + \dot{m}_2), \quad (2.6)$$

where γ_e and W_e are the specific-heat ratio and the molecular weight of the exhaust, p_r is the reactor-exit stagnation pressure, T_r is the reactor-exit stagnation temperature (given), p_a is the atmospheric pressure (a function of altitude), and R is the universal gas constant.[3] The specific-heat ratio and the molecular weight of the exhaust are given by

$$\gamma_e = (\gamma_1\dot{m}_1/(\gamma_1 - 1)W_1 + \gamma_2\dot{m}_2/(\gamma_2 - 1)W_2)/(\dot{m}_1/(\gamma_1 - 1)W_1 +$$

$$\dot{m}_2/(\gamma_2 - 1)W_2), \quad (2.7)$$

$$W_e = (\dot{m}_1 + \dot{m}_2)/(\dot{m}_1/W_1 + \dot{m}_2W_2), \quad (2.8)$$

where γ_1 and γ_2 are the specific-heat ratios, and W_1 and W_2 are the molecular weights, of the two propellants (after dissociation, if it occurs.)

The propellant flow rates satisfy the equations

$$\dot{m}_1 + \dot{m}_2 = -(2/(\gamma_e + 1))^{(\gamma_e + 1)/2(\gamma_e - 1)} (\gamma_e W_e / RT_r)^{1/2} p_r A_n, \qquad (2.9)$$

$$P = -(\dot{m}_1/\rho_1 + \dot{m}_2/\rho_2) p_r, \qquad (2.10)$$

where A_n is the nozzle throat area per unit initial mass (given), P is the effective pump power per unit initial mass (given), and ρ_1 and ρ_2 are the densities of the two propellants.[4] Substituting Eq. (2.7) and (2.8) into Eq. (2.9), and eliminating one of the flow rates between the resulting equation and Eq. (2.10), one obtains an equation that can be solved numerically for the other flow rate, for given reactor pressure. Thus $F = F(p_r, p_a(r))$; p_r is a control variable and r is a state variable.

The reactor pressure, p_r, must satisfy two inequality constraints. Since $\dot{m}_1$ and $\dot{m}_2$ are both negative, p_r must be bounded by the value p_A for which $\dot{m}_2 = 0$ (pure ammonia), and the value p_H for which $\dot{m}_1 = 0$ (pure hydrogen). To find p_A, set $\dot{m}_2 = 0$ in equations (2.9) and (2.10). After eliminating $\dot{m}_1$ between these equations, the resulting equation will be solved for p_r. This value will be p_A. Similarly, to find p_H, set $\dot{m}_1 = 0$ in equations (2.9) and (2.10). After eliminating $\dot{m}_2$ between these equations, the resulting equation will again be solved for p_r. This time this value will be p_H. Thus the constraints on the control vector are

$$p_H \leq u_1 = p_r \leq p_A, \qquad (2.11)$$

and

$$0 \le u_2 = \beta < \pi/2. \tag{2.12}$$

The drag and lift are given by

$$D = \gamma_a p_a M^2 C_D(M,\alpha) A_b/2, \tag{2.13}$$

$$L = \gamma_a p_a M^2 C_L(M,\alpha) A_b/2, \tag{2.14}$$

where γ_a is the specific-heat ratio of air, M is the flight Mach number, C_D and C_L are the drag and lift coefficients, α is the angle of attack, and A_b is the base area per unit initial mass (given). The Mach number and the angle of attack are given by

$$M = (W_a/\gamma_a R T_a)^{1/2}(\dot{r}^2 + (\dot{\phi} - \omega)^2 r^2)^{1/2}, \tag{2.15}$$

$$\alpha = \beta - \tan^{-1}(\dot{r}/(\dot{\phi} - \omega)r), \tag{2.16}$$

where W_a is the molecular weight of air, T_a is the atmospheric temperature (a function of altitude), and ω is the planet's rotational angular velocity. The temperature in the i-th layer of the atmosphere is given by

$$T_a = T_i + (r - r_i)\tau_i, \tag{2.17}$$

and the pressure by

$$p_a = p_i(T_a/T_i)^{-g_0 W_a/R\tau_i} \qquad (\tau_i \neq 0), \qquad (2.18)$$

$$p_a = p_i \exp(-(r - r_i)g_0 W_a/RT_i) \qquad (\tau_i = 0), \qquad (2.19)$$

where τ_i is a constant, and T_i and p_i are the temperature and pressure
at the base of the layer, at $r = r_i$. Thus drag and lift are functions
of the state variables and β, one of the control variables.

Thus the right-hand sides of the state differential equations are
functions of the state and control variables and the independent variable.

The initial and final state variables are given, i.e.

$$x_i(m_f) = x_{f_i}, \quad x_i(1) = x_{0_i}, \qquad (2.20)$$

where x_{0_i} and x_{f_i} are specified. In the case of a circular orbit,
$\dot{r}_f = 0$, and $\dot{\phi}_f = (g_0 r_0^2/r_f^3)^{1/2}$ where $\dot{r}_f$ and $\dot{\phi}_f$ are the final values of $\dot{r}$
and $\dot{\phi}$. The final conditions on the state variables can be written in the
form $G_i(x(m_f),m_f) = 0$, where $G_i(x(m),m) = x_i(m) - x_{f_i}$ for the x_{f_i} are
given.

Having formulated a mathematical model of the system and having
established initial and final conditions on the state variables, one must
define a performance index which is to be maximized or minimized. In
this case, the performance index is $I = m_f$ and this performance index is
to be maximized.

Thus the problem is in the form of an optimal control problem, where
one maximizes the functional $I = m_f$ subject to the differential con-
straints given by equations (2.3) - (2.5), the inequality constraints
given by equations (2.11) and (2.12), and the boundary conditions given
by equations (2.20).

CHAPTER III

THE OPTIMAL CONTROL PROBLEM

The control optimization problem is that of finding a control vector that "optimizes" the performance of the system. The optimal control problem formulated in Chapter II is a member of a class of optimal control problems considered below. In the general problem discussed below, the independent variable is chosen as t, for convenience, which is equivalent to the mass variable, m, of the problem discussed in Chapter II.

Let the state vector x be an N-vector and the control vector u be an M-vector; let x satisfy the set of first-order ordinary differential equations

$$dx_i/dt = f_i(x,u,t) \qquad i=1,\ldots,N, \tag{3.1}$$

and let the performance index be

$$I = P(x(t_f),t_f) + \int_{t_0}^{t_f} Q(x,u,t)dt. \tag{3.2}$$

I is said to be a functional of u. Let t_0 and $x(t_0)$ be given, and let

$$G_i(x(t_f),t_f) = 0 \qquad i=1,\ldots,L \ (L \leq N+1), \tag{3.3}$$

where t_f may or may not be given. The control vector u, may have to satisfy inequality constraints as well. The problem is to compute u(t), and t_f if it is not given, such that I is extremized.

This is the Bolza problem. If $P(x,t) = 0$, one has the Lagrange problem; if $Q(x,u,t) = 0$, one has the Mayer problem. The Bolza problem can be put into either the Lagrange or the Mayer form, if desired. The problem considered in this thesis is a Mayer problem, since $I = t_f$.

The following are necessary conditions for the extremization of the performance index:[4]

1. λ is introduced as a costate vector with N components, which must satisfy the set of first-order ordinary differential equations

$$d\lambda_i/dt = -\partial Q(x,u,t)/\partial x_i - \sum_{j=1}^{N} \lambda_j \partial f_j(x,u,t)/\partial x_i \qquad (3.4)$$

$i=1,\ldots,N,$

These equations are referred to as the costate differential equations.

2. The control vector u must satisfy the set of algebraic equations

$$\partial Q(x,u,t)/\partial u_i + \sum_{j=1}^{N} \lambda_j \partial f_j(x,u,t)/\partial u_i = 0 \qquad (3.5)$$

$i=1,\ldots,M,$

if there are no constraints on the control vector. These equations are referred to as the optimality conditions. If there are constraints on the control vector, the solution to these equations may violate these constraints. In that case, one chooses the control vector that comes as close as possible to

extremizing the function $Q(x,u,t) + \sum_{j=1}^{N} \lambda_j f_j(x,u,t)$ without violating the constraints.

3. The final conditions

$$\lambda_i(t_f) - \partial P(x(t_f),t_f)/\partial x_i + \sum_{j=1}^{L} q_j \partial G_j(x(t_f),t_f)/\partial x_i = 0 \qquad (3.6)$$

$i=1,\ldots,N,$

must be satisfied, where the q's are additional variables introduced which are referred to as Lagrange multipliers.

4. If t_f is not given, an additional final condition

$$Q(x(t_f),u(t_f),t_f) + \sum_{i=1}^{N} \lambda_i(t_f) f_i(x(t_f),u(t_f),t_f) +$$

$$\partial P(x(t_f),t_f)/\partial t - \sum_{i=1}^{L} q_i \partial G_i(x(t_f),t_f)/\partial t = 0, \qquad (3.7)$$

must be satisfied. This is referred to as the transversality condition.

Equations (3.1) and (3.4) can be solved for $x(t)$ and $\lambda(t)$, with the control vector determined by equation (3.5), and with the constants of integration, the Lagrange multipliers, and, if it is not given, t_f, determined by the initial conditions on the state vector, equations (3.3), and (3.6), and, if t_f is not given, equation (3.7).

The unknowns thus to be determined are an N-vector function x, an N-vector function λ, an M-vector function u, L q's, N constants of integration, and perhaps t_f. The equations from which these are to be determined consist of N state differential equations, N costate differential

equations, M optimality conditions, L terminal constraints, N final conditions on the costate variables, and if t_f is not given, a transversality condition. Thus there are the same number of equations as there are unknowns.

CHAPTER IV

THE NUMERICAL SOLUTION

In general, the control optimization problem leads to a nonlinear two-point boundary-value problem that cannot be solved analytically to obtain the optimal control function.

There are several methods for the numerical solution of the Bolza problem. The purpose here will be to present three iterative numerical techniques that could be used to solve the trajectory optimization problem. Two of the three techniques are procedures for solving general nonlinear two-point boundary-value problems. The third technique is applicable only to control optimization problems. The methods are briefly discussed below without the inequality constraints on the control vector.

Assuming that t_f is fixed, $x(t_f)$ is free, $x(t_0)$ is fixed, and t_0 is given, the two-point boundary-value problem can be summarized by the following equations:

$$dx_i/dt = f_i(x,u,t) \qquad i=1,\ldots,N, \tag{4.1}$$

$$d\lambda_i/dt = -\partial Q(x,u,t)/\partial x_i - \sum_{j=1}^{N} \lambda_j \partial f_j(x,u,t)/\partial x_i \qquad i=1,\ldots,N, \tag{4.2}$$

$$\partial Q(x,u,t)/\partial u_i + \sum_{j=1}^{N} \lambda_j \partial f_j(x,u,t)/\partial u_i = 0 \qquad i=1,\ldots,M, \tag{4.3}$$

$$x(t_0) = x_0, \tag{4.4}$$

$$\lambda_i(t_f) - \partial P(x(t_f),t_f)/\partial x_i = 0 \qquad i=1,\ldots,N. \tag{4.5}$$

The boundary conditions come from the N initial conditions on x, given by (4.4), and the N final conditions on λ, given by (4.5). The performance index is

$$I = P(x(t_f),t_f) + \int_{t_0}^{t_f} Q(x,u,t)dt.$$

Each of these numerical techniques are based on the following procedure:

An initial guess is used to obtain the solution to a problem in which one or more of the necessary conditions is not satisfied. This solution is then used to adjust the initial guess in an attempt to make the next solution come closer to satisfying all of the necessary conditions. If these steps are repeated and the iterative procedure converges, all the necessary conditions may eventually be satisfied.

In the steepest descent method,[5] the control function u(t) is guessed. To select a u(t) to begin the procedure , one utilizes whatever physical insight he has about the problem. Since the initial conditions on the state vector are given, the state differential equations can be numerically integrated forward to obtain x(t). Then λ(t_f) can be found by equation (4.5). Now since the final conditions on the costate vector are known, the costate differential equations (4.2), can be numerically integrated backwards to obtain λ(t). The next step is to solve the optimality equations (4.3), for a new u(t), using the calculated x(t) and λ(t). The procedure is repeated until the difference between successive anproximations becomes relatively small or until it is evident that convergence will not occur, in which case another u(t) needs to be selected.

The method of steepest descent is generally characterized by ease

of starting; the initial guess of u(t) is not usually crucial. Yet as a minimum is approached, the method has a tendency to converge slowly. In each iteration, numerical integration of 2N first-order differential equations is required. In addition, the optimality equations must be solved for a suitable number of values of t.

The variation of extremals technique[6] is an application of the Newton-Raphson iteration. This technique, unlike the steepest descent method, can be used to solve the general two-point boundary-value problem. In the variation of extremals technique, every trajectory generated by the algorithm satisfies equations (4.1) - (4.4), and hence is an extremal. Here an initial costate vector $\lambda(t_0)$ is assumed. Now the state and costate differential equations can be numerically integrated forwards to solve for $x(t)$ and $\lambda(t)$, with $u(t)$ being determined from the optimality condition. The calculated $\lambda(t_f)$ is a function of the assumed initial costate vector; it does not satisfy equation (4.5). Thus the set of N algebraic equations represented by equation (4.5) is solved for $\lambda_1(t_0),\dots,\lambda_N(t_0)$ by the Newton-Raphson method. This set of equations are solved for the corrections to the approximate solution. The computation of the right-hand sides of the linearized algebraic equations requires the numerical integration of the state and the costate differential equations. The computation of each column of the coefficient matrix requires another integration of the differential equations. Thus at each iteration the 2N differential equations must be integrated N+1 times and a set of N linear algebraic equations must be solved.

In the variational of extremals technique, divergence may result from a poor guess, but once convergence begins (if it does), it is rapid.

Actually, it may be better to guess $x(t_f)$ instead, since one would

probably have more knowledge about the final values of the states from the physical nature of the problem. If one guesses $x(t_f)$ instead of $\lambda(t_0)$, then the costate and state differential equations would now be integrated backwards in time.

The last technique to be considered is quasilinearization.[7] This method can also be used to solve the general two-point boundary-value problem. In this method, an $x(t)$ and $\lambda(t)$ are assumed. The state and costate differential equations are linearized about this nominal trajectory, bearing in mind that u, as a function of x, λ, and t, can be computed by solving the optimality conditions. One can obtain a general solution to the linearized state and costate differential equations in the following way. First, a particular solution is obtained by numerically integrating the linearized equations with the specified initial conditions on the state variables and with zero initial values for the costate variables. N linearly independent solutions to the corresponding homogeneous differential equations are obtained by solving the homogeneous equations N times, using a different set of initial conditions each time. A convenient choice of initial conditions would make one of the costate variables one, and all of the other variables zero. The general solution can be written as the sum of a particular solution and a linear combination of N complementary solutions--N rather than 2N, because the particular solution satisfies the initial conditions on the state variables by itself and the complementary solutions all satisfy homogeneous initial conditions on the state variables. The coefficients can be determined such as to satisfy the final conditions on the costate variables by solving a set of linear algebraic equations. The solution does not satisfy the exact state and costate differential equations. It satisfies a linear

approximation to those equations. This procedure is repeated using the new $x(t)$ and $\lambda(t)$ each time until the difference between successive approximations becomes sufficiently small or until it becomes evident that convergence will not occur, in which case another $x(t)$ and $\lambda(t)$ needs to be selected.

In the quasilinearization method, divergence may result from a poor guess. The initial guess of the state-costate trajectory is made primarily on the basis of whatever physical information is available about the particular problem being solved. Each iteration requires the solution of a set of 2N nonhomogeneous first-order linear differential equations, the solution of a set of 2N homogeneous linear first-order differential equations, with N different sets of initial conditions, and the solution of a set of N linear algebraic equations.

The three methods discussed thus far would all require modifications to handle the fixed final-state problem where the final value of the independent variable is not given and also to include inequality constraints whenever needed.[8]

A more flexible method is that of finite differences which could accomodate terminal constraints of any form, from the no-constraint problem to a fully specified state for free or fixed t_f. Replacement of the derivatives in the differential equations to be solved with finite-difference approximations results in a set of nonlinear algebraic equations to be solved. These equations will be solved with the Newton-Raphson iteration. The following discussion describes the procedure.

Let

$$h = (t_f - t_0)/K, \tag{4.6}$$

where K is a specified positive integer, and let

$$g^j = g(t_0 + jh), \tag{4.7}$$

where $g = x$, u, or λ.

Using the forward-, backward-, and central-difference approximations[9]

$$df(t)/dt = (-3f(t) + 4f(t + h) - f(t + 2h))/2h + 0(h^2), \tag{4.8}$$

$$df(t)/dt = (3f(t) - 4f(t - h) + f(t - 2h))/2h + 0(h^2), \tag{4.9}$$

$$df(t)/dt = (f(t + h) - f(t - h))/2h + 0(h^2), \tag{4.10}$$

the differential equations (3.1) can be approximated by the set of algebraic equations

$$(x_i^{j+1} - x_i^{j-1})/2h - f_i(x^j, u^j, t_0+jh) = 0 \tag{4.11}$$

$i=1,\ldots,N \qquad j=1,\ldots,K-1$,

and the differential equations (3.4) can be approximated by the set of algebraic equations

$$(-3\lambda_i^0 + 4\lambda_i^1 - \lambda_i^2)/2h + \partial Q(x^0, u^0, t_0)/\partial x_i + \sum_{j=1}^{N} \lambda_j \partial f_j(x^0, u^0, t_0)/\partial x_i = 0 \tag{4.12}$$

$i=1,\ldots,N$,

$$(\lambda_i^{j+1} - \lambda_i^{j-1})/2h + \partial Q(x^j, u^j, t_0 + jh)/\partial x_i +$$

$$\sum_{k=1}^{N} \lambda_k^j \partial f_k(x^j, u^j, t_0 + jh)/\partial x_i = 0 \qquad i=1,\ldots,N \quad j=1,\ldots,K-1, \qquad (4.13)$$

$$(3\lambda_i^K - 4\lambda_i^{K-1} + \lambda_i^{K-2})/2h + \partial Q(x^K, u^K, t_0 + Kh)/\partial x_i +$$

$$\sum_{j=1}^{N} \lambda_j^K \partial f_j(x^K, u^K, t_0 + Kh)/\partial x_i = 0 \qquad i=1,\ldots,N. \qquad (4.14)$$

The central-difference formula has been used from t between t_0 and t_f, the forward-difference formula for $t=t_0$ and the backward-difference formula for $t=t_f$. The equations represented by (4.11) are referred to as the state algebraic equations and equations (4.12) - (4.14) are referred to as the costate algebraic equations.

In order to conform to the same notation, equations (3.3), (3.5), (3.6), and (3.7), can be written as

$$G_i(x^K, t_0 + Kh) = 0 \qquad i=1,\ldots,L, \qquad (4.15)$$

$$\partial Q(x^j, u^j, t_0 + jh)/\partial u_i + \sum_{k=1}^{N} \lambda_k^j \partial f_k(x^j, u^j, t_0 + jh)/\partial u_i = 0 \qquad (4.16)$$

$$i=1,\ldots,M \qquad j=0,\ldots,K,$$

$$\lambda_i^K - \partial P(x^K, t_0 + Kh)/\partial x_i + \sum_{j=1}^{N} q_j \partial G_j(x^K, t_0 + Kh)/\partial x_i = 0 \qquad i=1,\ldots,N, \qquad (4.17)$$

$$Q(x^K, u^K, t_0 + Kh) + \sum_{i=1}^{N} \lambda_i^K f_i(x^K, u^K, t_0 + Kh) +$$

$$\partial P(x^K, t_0 + Kh)/\partial t - \sum_{i=1}^{N} q^i \partial G_i(x^K, t_0 + Kh)/\partial t = 0. \qquad (4.18)$$

The Newton-Raphson method may be used to solve equations (4.11) – (4.17), and if t_f is not given, equation (4.18), for the x's, u's, λ's, q's, and if t_f is not given, h. If there are inequality constraints on the control variables, one can compute the trajectory in segments. On some segments, the optimality conditions do not violate any of the constraints, and may therefore be ignored. On other segments, some of the optimality conditions cannot be satisfied without violating some of the constraints. On these segments, the optimality conditions that cannot be satisfied without violating the constraints are replaced by those constraints, treated as inequalities.

A particularly important problem is that of extremizing t_f for given final state. This problem may be treated as a Mayer problem with $P = t_f$, $Q = 0$, and

$$G_i(x(t_f), t_f) = x_i(t_f) - x_{f_i} = 0 \qquad i = 1, \ldots, N,$$

where x_{f_i} is given. In this case, equations (4.12) – (4.14) become

$$(-3\lambda_i^0 + 4\lambda_i^1 - \lambda_i^2)/2h + \sum_{j=1}^{N} \lambda_j^0 \partial f_j(x^0, u^0, t_0)/\partial x_i = 0 \qquad i = 1, \ldots, N, \qquad (4.19)$$

$$(\lambda_i^{j+1} - \lambda_i^{j-1})/2h + \sum_{j=1}^{N} \lambda_k^j \partial f_k(x^j, u^j, t_0 + jh)/\partial x_i = 0 \qquad (4.20)$$

$$i = 1, \ldots, N \qquad j = 1, \ldots, K-1,$$

$$(3\lambda_i^K - 4\lambda_i^{K-1} + \lambda_i^{K-2})/2h + \sum_{j=1}^{N} \lambda_j^K \partial f_j(x^K, u^K, t_0 + Kh)/\partial x_i = 0 \qquad (4.21)$$

$$i = 1, \ldots, N,$$

$$\sum_{k=1}^{N} \lambda_k^j \partial f_k(x^j, u^j, t_0 + jh)/\partial u_i = 0 \qquad i=1,\ldots,M \qquad j=0,\ldots,K, \qquad (4.22)$$

$$\sum_{i=1}^{N} \lambda_i^K f_i(x^K, u^K, t_0 + Kh) + 1 = 0. \qquad (4.23)$$

If necessary, central-difference approximations can be used for the derivatives in equations (4.19) - (4.22). Equations (4.11) and (4.19) - (4.23) can be solved by the Newton-Raphson method for the x's, u's, λ's, and h. Again, if there are inequality constraints on the control variables, one can divide the trajectory into segments on some of which the constraints may be ignored and on some of which some or all of the optimality conditions are replaced by constraints.

CHAPTER V

APPROXIMATE COMPUTATION OF THE OPTIMAL TRAJECTORY

In order to ensure convergence, it may be necessary to have a reasonable good initial approximation for the iterative procedure described in the previous chapter. Approximate trajectories can be obtained in the following way.

It is assumed that the effects of atmosphere are ignored, i.e., $p_a = 0$, and the specific heat ratios of the propellants are taken to be equal, i.e., $\gamma_1 = \gamma_2 = \gamma_e$. Substituting equation (2.9) into equation (2.6), one obtains $F = C_1 p_r$, where

$$C_1 = \gamma (2/(\gamma - 1))^{1/2} (2/(\gamma + 1))^{(\gamma+1)/2(\gamma-1)} A_n. \tag{5.1}$$

Thus the state differential equations take the simpler form

$$dr/dm = \dot{r}/\dot{m}, \tag{5.2}$$

$$d\dot{r}/dm = (-g_0 r_0/r^2 + r\dot{\phi}^2 + C_1 p_r \sin \beta/m)/\dot{m}, \tag{5.3}$$

$$d\dot{\phi}/dm = (-2\dot{r}\dot{\phi} + C_1 p_r \cos \beta/m)/\dot{m}r, \tag{5.4}$$

where $-\dot{m}$ is the total propellant flow rate per unit initial mass.

Next it is desired to compute $\dot{m}$ as a function of p_r. Letting $\dot{m}_1 + \dot{m}_2 = \dot{m}$ and $\gamma_e = \gamma$, and substituting equation (2.8) for W_e into (2.11), equation (2.11) now is of the form $f(\dot{m}, \dot{m}_1) = 0$. Letting $\dot{m}_2 = \dot{m} - \dot{m}_1$ in equation (2.12) and eliminating $\dot{m}_1$ between equations (2.11) and (2.12),

26

and equating the two of them, one has an equation in $\dot{m}$ only. This equation is a quadratic. Thus two solutions exist--one positive and one negative. The negative solution is used since $\dot{m}$ is negative by definition. Thus

$$\dot{m} = C_3/2C_2p_r - (C_3^2/4C_2^2p_r^2 + C_4p_r^2/C_2)^{1/2}, \tag{5.5}$$

where C_2, C_3, and C_4 are defined by

$$C_2 = \rho_1 W_2 - \rho_2 W_1, \tag{5.6}$$

$$C_3 = P\rho_1\rho_2(W_1 - W_2), \tag{5.7}$$

$$C_4 = \gamma(2/(\gamma + 1))^{(\gamma+1)/(\gamma-1)}(\rho_1 - \rho_2)W_1 W_2 A_n^2/RT_r, \tag{5.8}$$

where γ is the propellant specific heat ratio.

The costate differential equations become

$$d\lambda_1/dm = (-\lambda_2(2g_0 r_0^2/r^3 + \dot{\phi}^2) + \lambda_3(-2\dot{r}\dot{\phi} + C_1 p_r \cos\beta/m)/r^2)/\dot{m}, \tag{5.9}$$

$$d\lambda_2/dm = (-\lambda_1 + 2\lambda_3\dot{\phi}/r)/\dot{m}, \tag{5.10}$$

$$d\lambda_3/dm = (-2\lambda_2 r\dot{\phi} + 2\lambda_3\dot{r}/r)/\dot{m}. \tag{5.11}$$

The optimality conditions become

$$(\lambda_1 r + \lambda_2(-g_0 r_0^2/r^2 + r\dot{\phi}^2) - 2\lambda_3\dot{r}\dot{\phi}/r)(2C^3 + (C_3^2 + 4C_2C_4p_r^4)^{1/2}) +$$

$$(2C_1C_3p_r/m)(\lambda_2 \sin \beta + \lambda_3 \cos \beta/r) = 0, \tag{5.12}$$

$$\tan \beta - \lambda_2 r/\lambda_3 = 0. \tag{5.13}$$

β can be obtained from Eq. (5.13). Equation (5.12) can be rewritten as
a fourth-order polynomial in p_r, and this equation can be solved itera-
tively by the Bairstow method,[10] which is an application of the Newton-
Raphson method; one hopes that one at most of these solutions will be
between p_A and p_H. Thus the control variables can be calculated for
given values of the state and costate variables and the independent vari-
able.

For a set of estimated initial values of the costate variables, the
state and costate differential equations can be integrated numerically.
A good integration formula to use to do this is the Runge-Kutta method.
The right-hand sides of the differential equations (5.9) - (5.11) are
linear homogeneous functions in terms of λ_1, λ_2, and λ_3. Also the opti-
mality condition (5.12) is a linear homogeneous algebraic equation in λ_1,
λ_2, and λ_3. Equation (5.13) is also linear homogeneous in λ_2 and λ_3,
after multiplying it by λ_3. Hence, if the initial values of the costate
variables are multiplied by a common factor, the trajectory is unaffected,
and the subsequent values of the costate variables are all multiplied by
the same factor. Therefore one can assign an arbitrary initial value to
one of the costate variables, say λ_3. In order to satisfy the trans-
versality condition for any desired final mass, one needs only to multi-
ply the estimated initial values of the costate variables by the proper
factor. Thus any point on the trajectory can be a final state. A tra-
jectory computed in this way is a minimum-propellant trajectory for the

computed final state; but this final state is not known beforehand.
(All this is still true if one does not ignore the atmosphere and if one
does not assume that the two specific heat ratios are equal; the opti-
mality conditions and the right-hand sides of the costate differential
equations are still linear and homogeneous in the costate variables.)

It follows from the optimality condition on β, equation (5.13), that
λ_2 and λ_3 must always have the same sign since tan $\beta > 0$ for every point
on a realistic trajectory. Therefore, if λ_2 and λ_3 are ever going to
change signs, they must do it at the same time, i.e. $\lambda_2\lambda_3 > 0$. One way
to ensure that λ_2 and λ_3 never have opposite signs is to start λ_2 and λ_3
with the same signs and ensure that they both change in the same direc-
tion and that this direction is away from 0; obviously this is not a
necessary condition. If λ_3 changes in the direction away from 0, then
λ_3 and $d\lambda_3$ should have the same sign. Since dm is negative, one has
$\lambda_3 d\lambda_3/dm < 0$. By equation (5.11), this requires $\lambda_2/\lambda_3 < \dot{r}/r^2\dot{\phi}$. This
inequality implies the initial value of tan $\beta < \dot{r}/r\dot{\phi}$. Evidently at the
beginning of an optimized trajectory, one cannot have $\dot{r} = 0$, since β
must always be positive. Therefore, the trajectory must follow a non-
optimized boost phase which may use either chemical or nuclear propulsion.
(One suspects, although this has not been verified, that in order to
achieve a circular orbit, a shuttle would have to execute a nonoptimized
pushover maneuver.) Now to ensure that λ_2 and λ_3 change in the same
direction, one must have $(d\lambda_2/dm)(d\lambda_3/dm) > 0$. Using equations (5.10)
and (5.11) and assuming that $\lambda_2/\lambda_3 < \dot{r}/(r^2\dot{\phi})$, then in order to have
$(d\lambda_2/dm)(d\lambda_3/dm) > 0$, one must have $\lambda_1/\lambda_3 < 2\dot{\phi}/r$. One would also like to
have the initial values of λ_1 and λ_2 chosen such that p_r would be within
its limits p_H and p_A. The optimal p_r decreases during ascent. Therefore,

it may happen that at some point on the trajectory, the optimality condition will cease to have a solution between p_H and p_A. If this happens, one simply sets $p_r = p_H$.

CHAPTER VI

NUMERICAL RESULTS AND CONCLUSIONS

The purpose of this work is to determine the thrust vector program for a mixed-mode nuclear shuttle that minimizes the propellant required to reach a given orbit. The thrust is determined by the control variables β, which specifies the thrust direction, and p_r, which is the reactor pressure. Thrust is also a function of reactor pressure, reactor temperature, the nozzle throat area, and the turbopump power.

Application of the methods of optimal control theory to this problem results in a two-point boundary-value problem and an associated algebraic problem as has been shown in the earlier chapters. All of the numerical data presented in the following tables were computed by the methods described in Chapter V.

As will be soon demonstrated, the performance advantage of mixed-mode propulsion turned out to be quite small. Therefore, it was not worthwhile in this particular problem to compute more accurate trajectories using the rather expensive procedure described in Chapter IV. However, the correctness of the computer program, OPTCON, written to implement the latter procedure, was verified. In most cases of course, one will not be so fortunate as to obtain a simple approximate solution to an optimization problem, and the more elaborate procedure will have to be used from the beginning.

In Table 1, the state and control variables for a minimum-propellant trajectory, computed by the approximate method described in Chapter V, are given as functions of mass. The following set of input parameters are used:

$A_n = 10^{-3} \ m^2/ton,$

$P = 100 \ kw/ton,$

$T_r = 3000 \ K,$

$\gamma = 1.4.$

For these parameters, the reactor pressures for pure hydrogen and pure ammonia flow are:

$p_H = 6000 \ KN/m^2 \qquad p_A = 13,100 \ KN/m^2.$

The initial values for the costate variables are:

$\lambda_1(1) = 2.27 \times 10^{-11} \ m^{-1},$

$\lambda_2(1) = 1.69 \times 10^{-7} \ s/m,$

$\lambda_3(1) = 1 \ s.$

For reasons given in Chapter V, any point on this trajectory may be considered a final state. For example, the final mass for a minimum-propellant trajectory from radius $r = 6380$ km, radial velocity $\dot{r} = 0.5$ km/s, and angular velocity $\dot{\phi} = 7.25 \times 10^{-5} \ s^{-1}$ to $r = 6590$ km, $\dot{r} = 1.22$ km/s, and $\dot{\phi} = 4.99 \times 10^{-4} \ s^{-1}$, is $m_f = 0.6$.

As Table 1 indicates, the initial value of p_r is a little greater than p_H, but it drops to p_H immediately and remains there throughout the trajectory. The radial velocity, $\dot{r}$, decreases slightly just after ignition and then increases until cutoff. The angular velocity, $\dot{\phi}$, increases until just before cutoff. Then it decreases slightly.

Table 2 gives the state variables at $m = .2$ as functions of the initial value of the costate variable λ_1, denoted by $\lambda_1(1)$. All the input data are the same as for Table 1, except for the initial value of λ_1. As λ_1 increases, the final values of r and $\dot{r}$ decrease slightly and the final value of $\dot{\phi}$ increases slightly as well. As $\lambda_1(1)$ increases, the initial value of p_r decreases slightly.

Table 1. State and Control Variables as Functions of Mass for a Typical Trajectory

m	r, km	$\dot{r}$, km s^{-1}	$\dot{\phi}$, s^{-1}	p_r, kN m^{-2}	β
1.0	6380	0.500	0.725×10^{-4}	7210	0.823
0.9	6420	0.464	1.64	6000	0.835
0.8	6450	0.491	2.73	6000	0.873
0.7	6500	0.713	3.87	6000	0.935
0.6	6590	1.22	4.99	6000	1.02
0.5	6720	2.11	5.87	6000	1.13
0.4	6960	3.53	6.67	6000	1.24
0.3	7350	5.73	6.90	6000	1.36
0.2	7980	9.18	6.50	6000	1.48

Table 2. State Variables at m = .2 as Functions of the Initial Value of the Costate Variable λ_1

$\lambda_1(1)$, m^{-1}	r, km	$\dot{r}$, km s^{-1}	$\dot{\phi}$, s^{-1}
0.1×10^{-11}	8000	9.21	6.27×10^{-4}
0.2	8000	9.21	6.28
0.3	7990	9.21	6.29
0.4	7990	9.21	6.30
0.5	7990	9.21	6.31
0.6	7990	9.21	6.32
0.7	7990	9.20	6.33
0.8	7990	9.20	6.34
0.9	7990	9.20	6.35
1.0	7990	9.20	6.37
2.0	7980	9.18	6.47

Table 3 gives the state variables at m = .2 as a function of the initial value of the costate variable λ_2, denoted by $\lambda_2(1)$. All input parameters have the same values as those of Table 1, except for $\lambda_2(1)$. As $\lambda_2(1)$ increases, the final values of r and $\dot{r}$ increase, whereas the final value of $\dot{\phi}$ decreases. The initial value of p_r increases as $\lambda_2(1)$ increases, and for values of $\lambda_2(1) < 1.4 \times 10^{-7}$, the initial value of $p_r = p_H$. Therefore, the shuttle would be running strictly on hydrogen.

Table 4 shows the state variables at m = .2 as a function of A_n, the nozzle throat area. In this case, all the input parameters are the same as in Table 1, except for A_n. The final value of $\dot{r}$ and $\dot{\phi}$ increase as A_n increases while the final value of r first increases slightly, then decreases. For $A_n < 10^{-3}$ m^2/ton, the initial value of $p_r = p_H$. The pressure increases briefly after ignition, then drops back to p_H. As A_n increases above 10^{-3} m^2/ton, the initial value of p_r decreases and for $A_n > 1.2 \times 10^{-3}$ m^2/ton, this initial value = p_H.

Table 5 gives the state variables at m = .2 as functions of pump power per unit initial mass, P. As P increases, the final value of r first increases slightly, then decreases. The final value of r increases; for small values of P, it increases very rapidly. As P decreases below 100 kw/ton, the initial reactor pressure, p_r, decreases, and for P < 100 kw/ton, this initial value of $p_r = p_H$. The pressure increases briefly just after ignition, then falls back to p_H. As P increases above 100 kw/ton, the initial p_r decreases, until for P > 120 kw/ton, this initial pressure = p_H.

As is the case for any new idea, the feasability of mixed-mode propulsion must be demonstrated, both from a theoretical and a practical viewpoint. Even though mixed-mode propulsion is theoretically

Table 3. State Variables at m = .2 as Functions of the Initial Value of the Costate Variable λ_2

$\lambda_2(1)$, s m^{-1}	r, km	$\dot{r}$, km s^{-1}	$\dot{\phi}$, s^{-1}
1.4×10^{-7}	7820	8.95	7.52×10^{-4}
1.5	7880	9.04	7.10
1.6	7930	9.12	6.77

Table 4. State Variables at m = .2 as Functions of the Nozzle Throat Area, A_n

A_n, m^2 t^{-1}	r, km	$\dot{r}$, km s^{-1}	$\dot{\phi}$, s^{-1}
0.8×10^{-3}	7930	8.62	5.77×10^{-4}
0.9	7980	8.95	6.22
1.0	7980	9.18	6.50
1.1	7970	9.37	6.74
1.2	7960	9.53	7.04

Table 5. State Variables at m = .2 as Functions of the Pump Power, P

P, kw t^{-1}	r, km	$\dot{r}$, km s^{-1}	$\dot{\phi}$, s^{-1}
80	7930	8.62	5.77×10^{-4}
90	7980	8.95	6.22
100	7980	9.18	6.50
110	7970	9.37	6.74
120	7960	9.53	7.04

advantageous, it is not a good idea if it is not practical to implement or if it does not have substantial advantages over other propulsion systems. In designing a nuclear rocket, the first inclination is to choose hydrogen gas as the propellant because hydrogen has a low molecular weight--thus providing high exhaust velocity. Thus a comparison between mixed-mode propulsion and straight hydrogen propulsion is warranted here.

Using the same input parameters as in Table 1, but running only on hydrogen propellant, the cutoff radius r, is reduced from 7980 to 7960 km, the radial velocity $\dot{r}$, from 9.18 to 9.17 km/s, and the angular velocity $\dot{\phi}$, from 6.50×10^{-4} to 6.37×10^{-4} s^{-1}. Thus for a particular set of input parameters, it appears the use of mixed-mode propulsion only slightly improves the performance of a shuttle above pure hydrogen propulsion. This slight improvement would hardly justify the additional cost and complexity of mixed-mode propulsion.

A trajectory was computed for straight ammonia propellant and it was found that the performance for mixed-mode propulsion was far better than from ammonia propulsion.

It must be pointed out here that it may be possible to find input parameters for which the advantage of mixed-mode propulsion would be greater. For example, if the engines were very heavy, mixed-mode propulsion would be more advantageous. Chemical shuttles are marginal systems, whereas nuclear shuttles are not. Consequently, nuclear shuttles are much less sensitive to changes in mission and design parameters. This explains why the gains from mixed-mode propulsion are much larger for chemical shuttles than for nuclear shuttles.

In order to verify the correctness of OPTCON, the finite-difference program, a calculation was done for the unconstrained segment of the

trajectory using the same input data as in Table 1. The program, OPTCON, is of course a general one, and can be used for a wide variety of problems.

The results from OPTCON agreed closely with those computed from the method of Chapter V. Both methods were used to compute a trajectory with the initial state $r = 6381$ km, $\dot{r} = 0.5$ km/s, $\dot{\phi} = 7.249 \times 10^{-5}$ s^{-1}, and the final state $r = 6410$ km, $\dot{r} = 0.4691$ km/s, $\dot{\phi} = 1.511 \times 10^{-4}$ s^{-1}. The final mass as computed by OPTCON after five iterations was 0.9126, and the final mass as computed by the method of Chapter V was 0.9125. Midway through the OPTCON trajectory, p_r was 6,594 KN/m^2 and β was 0.8182. Midway through the other trajectory, p_r was 6,565 KN/m^2 and β was 0.8255. It appeared that after five iterations, the OPTCON solution was oscillating about a fixed point, but it was not apparent that the amplitude of the oscillations was decreasing.

CHAPTER VII

RECOMMENDATIONS

As was indicated previously, certain restrictions were imposed upon the problem from the outset, as was appropriate for an initial trajectory study. The following extensions to the study could be made, thus making the results more realistic. These studies will require more effort than the restricted problem. They are listed in order of increasing complexity.

1. Allowance for non-equatorial launch sites and nonequatorial orbits.

    Two more state variables, with the corresponding state and costate differential equations, and two more control variables, with the corresponding optimality conditions, must be introduced. The added state variables are latitude and the time derivative of latitude. Two additional angles must be given in order to specify the thrust direction.

2. Allowance for a coasting period; this is required in order to reach high orbits.

    Up until now, it has been assumed that the shuttle has been operating under power all the way. But high orbits cannot be reached under power all the way because the length of the powered trajectory cannot be long enough. In order to reach high orbit, one must incorporate a coasting phase in the trajectory. The following describes how to optimize a trajectory that includes a coasting phase. The first step is to compute an optimal trajectory with some specified final state. Next, the state vector

is computed after a specified coasting time. Finally, one computes an optimal trajectory from that state to the desired final state. The final mass is a function of the state at the beginning of the coasting period and the length of the coasting period. This function can be maximized by an application of the Newton-Raphson method. Thus one is faced with an optimal control problem inside a function optimization problem.

3. Allowance for non-ideal-gas effects in the nozzle flow; computations should be made for the limiting cases of equilibrium flow and frozen flow.[11]

Before now, thrust has been taken to be a function of reactor pressure, reactor temperature, the nozzle throat area, and the turbopump power, which it is. But thrust is really a more complicated function than was assumed before. It was assumed before that the molecular weight of the exhaust gas was a simple function of the mixture ratio and independent of pressure and temperature. It was also assumed that the specific heat of the exhaust gas is independent of temperature and pressure. Neither assumptions are correct. The molecular weight of the exhaust is a function of pressure and temperature and the specific heat is a function of temperature and pressure. Therefore, the calculation of thrust as a function of reactor pressure needs to be changed. There are two limiting cases which can both be handled easily, and which should both be considered. They are listed below.

1. The case of frozen flow

The composition of the exhaust gases is assumed to be

constant all along the nozzle at the reactor-exit equilibrium composition.

2. The case of equilibrium flow

    The gas is assumed to have its equilibrium composition at every point.

LIST OF REFERENCES

LIST OF REFERENCES

1. Salkeld, Robert, "Mixed-Mode Propulsion for the Space Shuttle," Astronautics & Aeronautics, August 1971, pp. 52-58.

2. Kingsbury, Donald, Technical Notes on Nuclear Rockets, Mathematics Department, McGill University, Montreal, 1976.

3. Hill, P. G. and Peterson, C. R., Mechanics and Thermodynamics of Propulsion, Addison-Wesley, Reading, Massachusetts, 1965, p. 356.

4. Kirk, Donald E., Optimal Control Theory;An Introduction, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1970, pp. 184-198.

5. Ibid., pp. 331-343.

6. Ibid., pp. 343-357.

7. Ibid., pp. 357-371.

8. Ibid., p. 372.

9. Hornbeck, Robert W., Numerical Methods, Quantum Publishers, Inc., New York, 1975, pp. 16-23.

10. Herriot, John G., Methods of Mathematical Analysis and Computation, John Wiley & Sons, Inc., New York, 1963, pp. 44-47.

11. Powell, Thomas Conley, Thermodynamics and Gas Dynamics of Rocket Motors (lecture notes), University of Tennessee Space Institute, Tullahoma, Tennessee, 1974.

APPENDICES

APPENDIX A

PROGRAM OPTCON

OPTCON solves the following problem: Let the state vector x be an N-vector and the control vector u be an M-vector; let x satisfy the set of first-order ordinary differential equations

$$dx_i/dt = F_i(x,u,t) \qquad i=1,\ldots,N . \tag{A.1}$$

Let t_0, $x(t_0)$, and $x(t_f)$ be given. The problem is to compute u(t) and t_f, such that t_f is extremized. This will be referred to as the extremal t_f problem.

The following are the necessary conditions for the extremization of t_f:

1. The N-vector λ must satisfy the set of first-order ordinary differential equations

$$d\lambda_i/dt = -\sum_{j=1}^{N} \lambda_j \partial F_j(x,u,t)/\partial x_i \qquad i=1,\ldots,N, \tag{A.2}$$

    λ is referred to as the costate vector, and these equations are referred to as the costate differential equations.

2. The control vector u must satisfy the set of algebraic equations

$$\sum_{j=1}^{N} \lambda_j \partial F_j(x,u,t)/\partial u_i = 0 \qquad i=1,\ldots,M, \tag{A.3}$$

    These equations are referred to as the optimality conditions.

3. Since t_f is not given, the final condition

$$\sum_{i=1}^{N} \lambda_i(t_f) F_i(x(t_f), u(t_f), t_f) + 1 = 0, \qquad (A.4)$$

must be satisfied. This is referred to as the transversality condition.

OPTCON uses a finite difference method to solve equations (A.1) – (A.4). The left-hand sides of equations (A.1) and (A.2) are replaced by central difference approximations. Thus equations (A.1) and (A.2) are approximated by a set of nonlinear algebraic equations; N state equations at each of K-2 points and N costate equations at each of K points. In addition, one has M optimality conditions at each of K points and a transversality condition. All of these equations are solved by the Newton-Raphson method for the N(K-2) state variables, the NK costate variables, the MK control variables, and t_f.

Initial estimates for the state, costate, and control variables must be supplied. To solve the set of nonlinear equations $F_1(x_1, \ldots, x_N) = 0, \ldots, F_N(x_1, \ldots, x_N) = 0$ by the Newton-Raphson method, the derivatives of the F's with respect to the x's must be calculated. This is done in subroutine JM by use of central difference formulas. In this case, subroutine G is used to compute the F's. In this case, the F's contain derivatives. Subroutine G calls D to calculate these derivatives by means of finite difference formulas. The right-hand sides of the state differential equations are calculated by DE. Subroutine DE would need to be rewritten for each new optimization problem; the only changes that would have to be made in the other subroutines are in DO loop indices and the like. NR calls subroutine GE to solve the linearized algebraic equations for the corrections to the unknowns.

A listing of the program follows.

```fortran
      IMPLICIT REAL*8 (A-H,O-Z)
C-----K=1+L*(M+N)+(L-2)*N
C-----DIMENSION A(K),A1(K),A2(K),DA(K),GA(K,K),U(L*M),X((L-2)*N),XO(N),
C-----XF(N),XL(L*N),Y(K),Y1(K),Y2(K)
      DIMENSION A(115),A1(115),A2(115),DA(115),GA(115,115),
     1U(30),X(39),XO(3),XF(3),XL(45),Y(115),Y1(115),Y2(115)
      DIMENSION P(8),R(8),T(8),TP(8)
      COMMON TO,XO,XF
      COMMON/COM1/AB,C1,C2,C3,C4,GO,OMEGA,P,R,RO,RU,T,TP,XK,XKA,XMA
      EXTERNAL G
      DATA TO,XO,XF/1.D3,6.381D6,5.D2,7.249D-5,6.410D6,
     14.691D2,1.511D-4/
      DATA AB,AN,GO,OMEGA,RO,RHO1,RHO2,RU,TR,W,XK,
     1XKA,XM1,XM2,XMA/1.D3,1.D0,9.807D0,7.272D-5,6.371D6,
     26.821D-1,7.D-2,8.314D3,3.D3,1.D5,2*1.4D0,8.516D0,
     332.016D0,2.896D1/
      DATA P/1.013D2,2.263D1,5.475D0,8.68D-1,1.109D-1,5.9D-2,
     11.821D-2,1.038D-3/, R/6.371D6,6.382D6,6.391D6,
     26.403D6,6.418D6,6.423D6,6.432D6,6.45D6/, T/
     332.8815D2,2.1665D2,2.1665D2,2.2865D2,2.7065D2,
     42.7065D2,2.5265D2,1.8065D2/, TP/-6.5D-3,0.D0,
     51.D-3,2.8D-3,0.D0,-2.D-3,-4.D-3,0.D0/
      DATA A(1)/-6.25D0/
      DATA XL/2.27D-11,1.69D-7,1.00D0,2.163D-11,1.69D-7,1.00D0,
     12.051D-11,1.69D-7,9.999D-1,1.935D-11,1.69D-7,9.998D-1,1.815D-11,
     21.691D-7,9.997D-1,1.691D-11,1.691D-7,9.995D-1,1.562D-11,1.692D-7,
     39.992D-1,1.429D-11,1.693D-7,9.989D-1,1.292D-11,1.694D-7,9.985D-1,
     41.149D-11,1.695D-7,9.980D-1,1.002D-11,1.696D-7,9.975D-1,
     58.502D-12,1.697D-7,9.969D-1,6.931D-12,1.699D-7,9.962D-1,
     65.307D-12,1.701D-7,9.953D-1,3.629D-12,1.703D-7,9.944D-1/
      DATA U/7.208D3,8.231D-1,7.103D3,8.233D-1,7.004D3,8.235D-1,
     16.909D3,8.237D-1,6.818D3,8.241D-1,6.730D3,8.245D-1,6.646D3,
     28.249D-1,6.565D3,8.255D-1,6.486D3,8.261D-1,6.41D3,8.269D-1,
     36.335D3,8.277D-1,6.264D3,8.286D-1,6.194D3,8.296D-1,6.125D3,
     48.308D-1,6.059D3,8.32D-1/
      DATA X/6.383D6,4.992D2,7.734D-5,6.385D6,4.982D2,8.231D-5,
     16.386D6,4.969D2,8.738D-5,6.388D6,4.954D2,9.258D-5,6.39D6,4.937D2,
     29.788D-5,6.392D6,4.917D2,1.033D-4,6.394D6,4.895D2,1.089D-4,
     36.397D6,4.871D2,1.145D-4,6.399D6,4.845D2,1.203D-4,6.401D6,
     44.818D2,1.262D-4,6.403D6,4.788D2,1.322D-4,6.406D6,4.757D2,
     51.384D-4,6.408D6,4.725D2,1.447D-4/
      DO 1 I=1,45
    1 XL(I)=XL(I)/9.384D-7
      DO 2 I=1,45
    2 A(I+1)=XL(I)
      DO 3 I=1,30
    3 A(I+46)=U(I)
      DO 4 I=1,39
    4 A(I+76)=X(I)
      CALL ASSIGN(6,'OPTCON.OUT')
      C1=XK*DSQRT(2./(XK-1.))*(2./(XK+1.))**(0.5*(XK+1.)/(XK-1.))*AN
```

```
       C2=RHO1*XM2-RHO2*XM1
       C3=W*RHO1*RHO2*(XM1-XM2)
       C4=(RHO1-RHO2)*XM1*XM2*AN**2*XK*(2./(XK+1.))**
      1((XK+1.)/(XK-1.))/(RU*TR)
C-----CALL NR(G,K,A,DA,GA,A1,A2,Y,Y1,Y2)
       CALL NR(G,115,A,DA,GA,A1,A2,Y,Y1,Y2)
       WRITE (6,10)
       WRITE (6,100) A(1)
       WRITE (6,20)
C-----WRITE (6,110) (A(I),I=2,1+L*N)
       WRITE (6,110) (A(I),I=2,46)
       WRITE (6,30)
C-----WRITE (6,120) (A(I),I=2+L*N,1+L*(M+N))
       WRITE (6,120) (A(I),I=47,76)
       WRITE (6,40)
C-----WRITE (6,110) (A(I),I=2+L*(M+N),K)
       WRITE (6,110) (A(I),I=77,115)
10     FORMAT (1X,1HH)
20     FORMAT (1X,1HP)
30     FORMAT (1X,1HU)
40     FORMAT (1X,1HX)
100    FORMAT (1X,1PD10.3)
C-----FORMAT (N(1X,1PD10.3))
110    FORMAT (3(1X,1PD10.3))
C-----FORMAT (M(1X,1PD10.3))
120    FORMAT (2(1X,1PD10.3))
       STOP
       END
       SUBROUTINE NR(F,N,X,DX,FX,X1,X2,Y,Y1,Y2)
C-----SUBROUTINE NR USES THE NEWTON-RAPHSON METHOD TO SOLVE A SET
C-----OF N NONLINEAR ALGEBRAIC EQUATIONS F(X)=0, WHERE F IS AN
C-----N-VECTOR FUNCTION OF GIVEN FORM, AND X IS AN UNKNOWN N-VECTOR.
C-----THE NEWTON-RAPHSON METHOD REPLACES A SET OF NONLINEAR ALGEBRAIC
C-----EQUATIONS WITH LINEAR EQUATIONS, THE SOLUTION OF WHICH SHOULD
C-----BE CLOSE TO THE ORIGINAL EQUATIONS.  SUBROUTINE NR CALLS JM
C-----WHICH USES A THREE-POINT CENTRAL DIFFERENCE FORMULA TO COMPUTE
C-----THE COEFFICIENT MATRIX(JACOBIAN MATRIX).  NR THEN CALLS
C-----SUBROUTINE GE WHICH USES THE GAUSS ELIMINATION METHOD TO SOLVE
C-----THE LINEARIZED EQUATIONS FOR THE CORRECTION VECTOR.  INITIALLY,
C-----X MUST CONTAIN AN ESTIMATE SOLUTION.  THIS IS OVERWRITTEN BY
C-----THE SUCCESSIVE APPROXIMATIONS TO THE SOLUTION.
C-----      THE USER SUPPLIES A SUBPROGRAM TO EVALUATE THE EQUATIONS
C-----THEMSELVES.
C-----      THE INPUT DATA ARE:
C-----N  NUMBER OF EQUATIONS=NUMBER OF UNKNOWNS
C-----X  VECTOR OF UNKNOWNS
C-----      THE WORKING ARRAYS, WITH THEIR DIMENSIONS, ARE:
C-----DX  N
C-----FX  N X N
C-----X1  N
```

```
C-----X2   N
C-----Y    N
C-----Y1   N
C-----Y2   N
C-----DX, FX, X, X1, X2, Y, Y1, AND Y2 MUST BE DIMENSIONED IN THE
C-----CALLING PROGRAM.
C-----     F IS A DUMMY NAME FOR A USER SUPPLIED SUBROUTINE THAT
C-----COMPUTES F(X).  THIS SUBROUTINE'S ARGUMENT LIST IS (X,Y); X IS
C-----AN N X 1 INPUT ARRAY, AND Y IS AN N X 1 OUTPUT ARRAY CONTAINING
C-----F(X).
C-----     THIS SUBROUTINE'S ACTUAL NAME MUST APPEAR IN AN EXTERNAL
C-----STATEMENT IN THE CALLING PROGRAM.  X1, X2, Y1, AND Y2 ARE N
C-----VECTORS.  X, DX, FX, X1, X2, Y1, AND Y2 MUST BE DIMENSIONED IN
C-----THE CALLING PROGRAM.
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION DX(N),FX(N,N),X(N),X1(N),X2(N),Y(N),Y1(N),
     1Y2(N)
      EXTERNAL F
1     CALL JM(F,N,X,FX,X1,X2,Y1,Y2)
      CALL F(X,Y)
      CALL GE(FX,Y,N,DX)
      DO 2 I=1,N
2     X(I)=X(I)-DX(I)
      WRITE (6,100) X(1)
      WRITE (6,110) (X(I),I=2,46)
      WRITE (6,120) (X(I),I=47,76)
      WRITE (6,110) (X(I),I=77,115)
100   FORMAT (1X,1PD10.3)
110   FORMAT (3(1X,1PD10.3))
120   FORMAT (2(1X,1PD10.3))
      DO 3 I=1,N
      IF (DABS(DX(I)/X(I)).GT.1.D-6) GO TO 1
3     CONTINUE
      RETURN
      END
      SUBROUTINE JM(F,N,X,FX,X1,X2,Y1,Y2)
C-----SUBROUTINE JM COMPUTES THE DERIVATIVES OF THE N-VECTOR FUNTION
C-----F(X) WHERE X IS AN N-VECTOR.  SUBROUTINE JM USES A THREE-POINT
C-----CENTRAL-DIFFERENCE FORMULA TO COMPUTE THE DERIVATIVES OF THE
C-----FUNCTIONS.
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION FX(N,N),X(N),X1(N),X2(N),Y1(N),Y2(N)
      DO 1 I=1,N
      X1(I)=X(I)
1     X2(I)=X(I)
      DO 3 J=1,N
      X1(J)=0.999*X(J)
      X2(J)=1.001*X(J)
      CALL F(X1,Y1)
      CALL F(X2,Y2)
      DO 2 I=1,N
```

```
2       FX(I,J)=(Y2(I)-Y1(I))/(0.002*X(J))
        X1(J)=X(J)
3       X2(J)=X(J)
        RETURN
        END
        SUBROUTINE GE(A,C,N,X)
C-----THE GAUSS ELIMATION SUBROUTINE SOLVES THE SET OF SIMULTANEOUS
C-----LINEAR ALGEBRAIC EQUATIONS AX=C WHERE A IS A GIVEN N X N
C-----MATRIX, C IS A GIVEN N-VECTOR, AND X IS AN UNKNOWN N-VECTOR.
C-----A, C, AND X MUST BE DIMENSIONED IN THE CALLING PROGRAM.   THE
C-----USER MUST IINPUT THE NUMBER OF EQUATIONS TO BE SOLVED.   THIS
C-----SUBROUTINE RETURNS THE SOLUTION OF THESE LINEAR ALGEBRAIC
C-----EQUATIONS TO THE MAIN PROGRAM.   THE COEFFICIENT MATRIX A AND
C-----THE C VECTOR ARE OVERWRITTEN DURING EXECUTION.
        IMPLICIT REAL*8 (A-H,O-Z)
        DIMENSION A(N,N),C(N),X(N)
        M=N-1
        DO 3 K=1,M
        L=K+1
        DO 2 I=L,N
        IF (DABS(A(I,K)).LE.DABS(A(K,K))) GO TO 2
        DO 1 J=K,N
        AIJ=A(I,J)
        AKJ=A(K,J)
        A(I,J)=AKJ
1       A(K,J)=AIJ
        CI=C(I)
        CK=C(K)
        C(I)=CK
        C(K)=CI
2       CONTINUE
        DO 3 I=L,N
        B=A(I,K)/A(K,K)
        C(I)=C(I)-B*C(K)
        DO 3 J=L,N
3       A(I,J)=A(I,J)-B*A(K,J)
        X(N)=C(N)/A(N,N)
        DO 5 I=1,M
        K=N-I
        L=K+1
        X(K)=C(K)
        DO 4 J=L,N
4       X(K)=X(K)-A(K,J)*X(J)
5       X(K)=X(K)/A(K,K)
        RETURN
        END
        SUBROUTINE DE(XDOT,XM,U,X)
C-----SUBROUTINE DE EVALUTATES THE RIGHT HAND SIDES OF THE STATE
C-----DIFFERENTIAL EQUATIONS FOR GIVEN VALUES OF X, U, AND T.
        IMPLICIT REAL*8 (A-H,O-Z)
        DIMENSION P(8),R(8),T(8),TP(8),U(2),X(3),XDOT(3)
```

```
          COMMON/COM1/AB,C1,C2,C3,C4,G0,OMEGA,P,R,R0,RU,T,TP,XK,XKA,XMA
          PR=U(1)
          BETA=U(2)
          Q=C3/(2.*C2*PR)-DSQRT((C3/(2.*C2*PR))**2+C4*PR**2/C2)
          I=0
          IF (X(1).LT.R(2)) I=1
          IF (I.EQ.1) GO TO 3
          IF (X(1).GT.R(8)) I=8
          IF (I.EQ.8) GO TO 3
          DO 2 I=2,7
          IF (X(1).GT.R(I).AND.X(1).LT.R(I+1)) GO TO 3
2         CONTINUE
3         TA=T(I)+(X(1)-R(I))*TP(I)
          IF (TP(I).EQ.0.) PA=P(I)*DEXP((R(I)-X(1))*G0*
         1XMA/(RU*T(I)))
          IF (TP(I).NE.0.) PA=P(I)*(T(I)/TA)**(G0*XMA/(RU*
         1TP(I)))
          PA=0.
          ALPHA=U(2)-DATAN(X(2)/(X(1)*(X(3)-OMEGA)))
          XNM=DSQRT(XMA*(X(2)**2+(X(1)*(X(3)-OMEGA))**2)/
         1(XKA*RU*TA))
          FX=C1*PR-XKA*PA*XNM**2*CD(ALPHA,XNM)*AB/2.
          FY=XKA*PA*XNM**2*CL(ALPHA,XNM)*AB/2.
          SBETA=DSIN(BETA)
          CBETA=DCOS(BETA)
          XDOT(1)=X(2)/Q
          XDOT(2)=(-G0*(R0/X(1))**2+X(1)*X(3)**2+(FX*SBETA+
         1FY*CBETA)/XM)/Q
          XDOT(3)=(-2.*X(2)*X(3)+(FX*CBETA-FY*SBETA)/XM)/
         1(Q*X(1))
          RETURN
          END
          SUBROUTINE D(FU,FX,T,U,X)
C-----SUBROUTINE D EVALUATES THE DERIVATIVES IN EQUATIONS 4.19-4.22
C-----BY MEANS OF CENTRAL DIFFERENCE FORMULAS.
          IMPLICIT REAL*8 (A-H,O-Z)
C-----DIMENSION F1(N),F2(N),FU(N,M),FX(N,N),U(M),U1(M),U2(M)
C-----X(N),X1(N),X2(N)
          DIMENSION F1(3),F2(3),FU(3,2),FX(3,3),U(2),U1(2),U2(2),
         1X(3),X1(3),X2(3)
C-----DO 1 I=1,M
          DO 1 I=1,2
          U1(I)=U(I)
1         U2(I)=U(I)
C-----DO 2 I=1,N
          DO 2 I=1,3
          X1(I)=X(I)
2         X2(I)=X(I)
C-----DO 4 J=1,M
          DO 4 J=1,2
          U1(J)=0.999*U(J)
```

```fortran
      U2(J)=1.001*U(J)
      CALL DE(F1,T,U1,X)
      CALL DE(F2,T,U2,X)
C-----DO 3 I=1,N
      DO 3 I=1,3
3     FU(I,J)=(F2(I)-F1(I))/(0.002*U(J))
      U1(J)=U(J)
4     U2(J)=U(J)
C-----DO 6 J=1,N
      DO 6 J=1,3
      X1(J)=0.999*X(J)
      X2(J)=1.001*X(J)
      CALL DE(F1,T,U,X1)
      CALL DE(F2,T,U,X2)
C-----DO 5 I=1,N
      DO 5 I=1,3
5     FX(I,J)=(F2(I)-F1(I))/(0.002*X(J))
      X1(J)=X(J)
6     X2(J)=X(J)
      RETURN
      END
      SUBROUTINE G(A,Y)
C-----SUBROUTINE G COMPUTES THE LEFT-HAND SIDES OF THE STATE,COSTATE,
C-----OPTIMALITY, AND TRANSVERSALITY EQUATIONS.  A USER-SUPPLIED
C-----SUBROUTINE DE, WHICH EVALUATES THE RIGHT-HAND SIDES OF THE STATE
C-----DIFFERENTIAL EQUATIONS, IS REQUIRED.  SUBROUTINE G ALSO USES ANOTHER
C-----SUBROUTINE, D, TO CALCULATE THE DERIVATIVES IN THESE EQUATIONS.
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON T0,X0,XF
C-----K=1+L*(M+N)+(L-2)*N
C-----DIMENSION A(K),F(N),FU(N,M),FX(N,N),GP(N,L),GU(M,L),GX(N,L-2),P(N,L),
C----1PJ(N),U(M,L),UJ(M),X(N,L),X0(N),XF(N),XJ(N),Y(K)
      DIMENSION A(115),F(3),FU(3,2),FX(3,3),
     1GP(3,15),GU(2,15),GX(3,13),P(3,15),PJ(3),U(2,15),UJ(2),X(3,15),
     2X0(3),XF(3),XJ(3),Y(115)
      H=A(1)
C-----DO 1 J=1,L
      DO 1 J=1,15
C-----DO 1 I=1,N
      DO 1 I=1,3
C-----P(I,J)=A(I+N*(J-1)+1)
1     P(I,J)=A(I+3*(J-1)+1)
C-----DO 2 J=1,L
      DO 2 J=1,15
C-----DO 2 I=1,M
      DO 2 I=1,2
C-----U(I,J)=A(I+M*(J-1)+L*N+1)
2     U(I,J)=A(I+2*(J-1)+46)
C-----DO 3 I=1,N
      DO 3 I=1,3
      X(I,1)=X0(I)
```

53

```fortran
C-----X(I,L)=XF(I)
3     X(I,15)=XF(I)
C-----DO 4 J=2,L-1
      DO 4 J=2,14
C-----DO 4 I=1,N
      DO 4 I=1,3
C-----X(I,J)=A(I+N*(J-2)+L*(M+N)+1)
4     X(I,J)=A(I+3*(J-2)+76)
C-----DO 13 J=1,L
      DO 13 J=1,15
C-----DO 5 I=1,M
      DO 5 I=1,2
5     UJ(I)=U(I,J)
C-----DO 6 I=1,N
      DO 6 I=1,3
      PJ(I)=P(I,J)
6     XJ(I)=X(I,J)
      TJ=T0+(J-1)*H
      CALL DE(F,TJ,UJ,XJ)
      CALL D(FU,FX,TJ,UJ,XJ)
C-----DO 7 I=1,M
      DO 7 I=1,2
      GU(I,J)=0.
C-----DO 7 K=1,N
      DO 7 K=1,3
7     GU(I,J)=GU(I,J)+P(K,J)*FU(K,I)
      IF (J.EQ.1) GO TO 9
C-----IF (J.EQ.L) GO TO 11
      IF (J.EQ.15) GO TO 11
C-----DO 8 I=1,N
      DO 8 I=1,3
      GX(I,J-1)=(X(I,J+1)-X(I,J-1))/(2.*H)-F(I)
      GP(I,J)=(P(I,J+1)-P(I,J-1))/(2.*H)
C-----DO 8 K=1,N
      DO 8 K=1,3
8     GP(I,J)=GP(I,J)+P(K,J)*FX(K,I)
      GO TO 13
C-----DO 10 I=1,N
9     DO 10 I=1,3
      GP(I,1)=(-3.*P(I,1)+4.*P(I,2)-P(I,3))/(2.*H)
C-----DO 10 K=1,N
      DO 10 K=1,3
10    GP(I,1)=GP(I,1)+P(K,1)*FX(K,I)
      GO TO 13
11    GH=1.
C-----DO 12 I=1,N
      DO 12 I=1,3
      GH=GH+P(I,J)*F(I)
      GP(I,J)=(3.*P(I,J)-4.*P(I,J-1)+P(I,J-2))/(2.*H)
C-----DO 12 K=1,N
      DO 12 K=1,3
```

```
12      GP(I,J)=GP(I,J)+P(K,J)*FX(K,I)
13      CONTINUE
        Y(1)=GH
C-----DO 15 J=1,L
        DO 15 J=1,15
C-----DO 14 I=1,N
        DO 14 I=1,3
C-----Y(I+N*(J-1)+1)=GP(I,J)
14      Y(I+3*(J-1)+1)=GP(I,J)
C-----DO 15 I=1,M
        DO 15 I=1,2
C-----Y(I+M*(J-1)+L*N+1)=GU(I,J)
15      Y(I+2*(J-1)+46)=GU(I,J)
C-----DO 16 J=1,L-2
        DO 16 J=1,13
C-----DO 16 I=1,N
        DO 16 I=1,3
C-----Y(I+N*(J-1)+L*(M+N)+1)=GX(I,J)
16      Y(I+3*(J-1)+76)=GX(I,J)
        RETURN
        END
        FUNCTION CD(ALPHA,XNM)
        IMPLICIT REAL*8 (A-H,O-Z)
        CD=0.
        RETURN
        END
        FUNCTION CL(ALPHA,XNM)
        IMPLICIT REAL*8 (A-H,O-Z)
        CL=0.
        RETURN
        END
```

APPENDIX B

NEWTON-RAPHSON METHOD

Consider the set of nonlinear algebraic equations

$$F_j(x_1,x_2,\ldots,x_n) = 0 \qquad j=1,\ldots,n.$$

Let $(x_1^i,x_2^i,\ldots,x_n^i)$ be an approximate solution. The Taylor series expansion of $F_j(x_1,x_2,\ldots,x_n)$ to first order is

$$F_j(x_1,x_2,\ldots,x_n) \simeq F_j(x_1^i,x_2^i,\ldots,x_n^i) + \sum_{k=1}^{n}(x_k-x_k^i)\partial F_j(x_1^i,x_2^i,\ldots,x_n^i)/\partial x_k$$

$$j=1,2,\ldots,n,$$

If $x_1^{i+1},x_2^{i+1},\ldots,x_n^{i+1}$ is then defined such that

$$F_j(x_1^i,x_2^i,\ldots,x_n^i) + \sum_{k=1}^{n}(x_k^{i+1} - x_k^i)\partial F_j(x_1,x_2,\ldots,x_n)/\partial x_k = 0 \qquad j=1,2,\ldots,n,$$

then x_1^{i+1}, $x_2^{i+1},\ldots,x_n^{i+1}$ may be an improved approximation. Thus a set of nonlinear algebraic equations has been replaced by a set of linear equations, the solution of which should be close to the solution of the original equations. This set of linear equations can be solved by the Gauss elimination method, and the values obtained can be used to obtain another approximation, and so on, until the difference between successive approximations becomes sufficiently small, or until it becomes clear that convergence will not occur. In the latter case, one starts over with a new initial estimate.

This procedure is called the Newton-Raphson method.

The n×n matrix having $\partial F_j/\partial x_k$ in the jth row and the kth column is called the Jacobian. It is often necessary to calculate the Jacobian numerically by using central-difference formulas.

In the case of a single equation,

$$x^{i+1} = x^i - F(x^i)/(dF(x^i)/dx).$$